On Efficient Sensor Scheduling for Linear Dynamical Systems

Michael P. Vitus, Wei Zhang, Alessandro Abate, Claire J. Tomlin,

Abstract— **Consider a set of sensors estimating the state of a process in which only a subset can operate at each time-step due to constraints on the overall system. Therefore, the problem is to choose which sensors should operate at each time-step to minimize a weighted function of the state estimate error covariance at each time-step. This work investigates developing tractable algorithms to solve for the optimal and suboptimal sensor schedule. First, a condition on when a partial sensor schedule cannot lead to the optimal schedule is developed. Second, using this condition, both an optimal and a suboptimal algorithm are devised to prune the search tree to enable the solution of larger systems and longer time horizons. The suboptimal algorithm trades-off between the quality of the solution and the complexity of the problem through a tuning parameter. Third, a hierarchical algorithm is formulated based on the suboptimal algorithm; it uses the results from a low complexity solution to further prune branches for a higher complexity solution which usual provides a better solution. Finally, numerical simulations are performed to demonstrate the performance of the proposed algorithms.**

I. INTRODUCTION

The problem of *sensor scheduling* is to select a subset of sensors to operate at each time-step that minimizes a weighted function of the estimation error at each time-step. Sensor scheduling is an essential technology for applications that have constraints such that only a subset of the sensors can operate at each time-step. An example of such a system is a wireless sensor network which comprises of multiple nodes monitoring an external process. The nodes perform some local processing of the data and then transmit it to a central aggregation process. Constraints on the network's communication bandwidth might not allow all of the nodes to communicate at each time-step. Also, each node may only have a limited amount of power and therefore it will turn off to conserve power when its measurement is not required. Consequently, the objective is to manage the schedule of nodes' measurements. Sensor scheduling can also be used to handle sensors which interfere with one another, as with sonar range-finding sensors, and thus cannot operate at the same time.

Meier III et al. [1] proposed a solution to the discrete time scheduling problem through the use of dynamic programming which enumerates all possible sensor schedules, but the curse of dimensionality makes this method intractable for longer schedule horizons. A local gradient method is also proposed which is more likely to be computationally feasible, but only provides a suboptimal solution. In [2], a relaxed dynamic programming procedure can be applied to obtain a suboptimal strategy, which is bounded by a pre-specified distance within optimality, that minimizes the trace of the final state estimate covariance. A convex optimization procedure was developed in [3] as a heuristic to solve the problem of selecting k sensors from a set of m. Although no optimality guarantees can be provided for the solution, numerical experiments suggest that it performs well. In [4], a sliding window algorithm and a thresholding algorithm are applied in order to prune the search tree, but only provide a suboptimal solution with no bound on the optimality. A method that switches sensors randomly according to a probability distribution to obtain the best upperbound on the expected steady-state performance is developed in [5]. A sensor scheduling algorithm which trades off the performance and sensor usage costs was devised in [6], and formulated as a partially observed Markov decision process and solved via an approximation process.

The sensor scheduling problem can also be thought of as the dual of the optimal control problem for switched systems, which form a class of hybrid systems. A switched system usually consists of a family of subsystems, each with specified dynamics, and allows for switching between the different subsystems at various times. The analysis and design of controllers for hybrid systems has received a large amount of attention from the research community [7], [8], [9], [10], [11], [12], [13], [14]. Specifically, Zhang et al. [13], [14] proposed a method based on dynamic programming to solve for the optimal discrete mode sequence and continuous input for the discrete-time linear quadratic regulation problem for switched linear systems. They proposed several efficient and computationally tractable algorithms for obtaining the optimal and bounded suboptimal solution through effective pruning of the search tree, which grows exponentially with the horizon length.

This work presents three main contributions that arise out of the insights from the control of switched systems in [13], [14]. First, conditions are presented that express when a partial sensor schedule is non-optimal. Second, based on the previous condition, two efficient pruning techniques are developed which provide the optimal or suboptimal solutions. These algorithms significantly prune the search tree enabling the solution of larger systems with longer schedule horizons than through the brute force enumeration. The suboptimal algorithm includes a tuning parameter which trades off the quality of the solution with the complexity of

Michael P. Vitus and Alessandro Abate are with the Department of Aeronautics and Astronautics, Stanford University, CA 94305. Email: vitus,aabate@stanford.edu

Wei Zhang is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN47906. Email: zhang70@purdue.edu

Claire J. Tomlin is with the Department of Electrical Engineering and Computer Sciences. Email: tomlin@eecs.berkeley.edu

the problem, for small and large values respectively. Third, a hierarchical algorithm is also formulated which reduces the complexity of the problem while maintaining the quality of the solution. This is achieved by using the results from larger parameters to prune branches for smaller parameters. Even though there is no bound on the quality of the suboptimal solution, numerical experiments suggest that the solution is usually within a few percent of the optimal value of the objective function.

The paper proceeds as follows. Section II describes the standard sensor scheduling problem formulation and presents a useful property of the Kalman filter. Then, properties of the objective function are explored and a theorem which is useful for pruning branches in the search tree is presented in Section III. In Section IV, a description of tractable algorithms for determining the optimal and suboptimal solutions are provided, and the performance of both algorithms are explored through random simulations. In Section V, the hierarchical method is formulated and the paper concludes with directions of future work.

II. Problem Formulation

Consider the following time-varying, linear stochastic system defined by,

$$x\left(k+1\right)=A\left(k\right)x\left(k\right)+w\left(k\right),\;\forall k\in T_N \tag{1}$$

where $x\left(k\right)\in\mathbb{R}^n$ is the state of the system, $w\left(k\right)\in\mathbb{R}^n$ is the process noise and $T_N=\{0,\dots,N-1\}$ is the horizon. The initial state, $x(0)$, is assumed to be distributed via a zero mean Gaussian distribution, $x(0)\sim\mathcal{N}\left(0,\Sigma_0\right)$. At each time step, only one sensor is allowed to operate from a set of M available sensors. The dynamics of the i^{th} sensor is given by,

$$y_i\left(k\right)=C_ix\left(k\right)+v_i\left(k\right),\;\forall k\in T_N \tag{2}$$

where $y_i(k)\in\mathbb{R}^p$ and $v_i(k)\in\mathbb{R}^p$ are the measurement output and measurement noise of the i^{th} sensor at time k, respectively. The process and measurement noise are zero mean Gaussian distributions,

$$w\left(k\right)\sim\mathcal{N}\left(0,\Sigma_w\right),\;v_i\left(k\right)\sim\mathcal{N}\left(0,\Sigma_{v_i}\right)\;\forall i\in\mathbb{M} \tag{3}$$

where $\mathbb{M}\triangleq\{1,\dots,M\}$ is the set of M sensors. The process noise, measurement noise and initial state are also assumed to be independent random variables. Denote by $\mathbb{M}^t$ the set of all ordered sequences of sensor schedules of length t where $t\leq N$. An element $\sigma^t\in\mathbb{M}^t$ is called a (t-horizon) *sensor schedule*. Under a given sensor schedule σ^t, the measurement sequence is determined by,

$$y(k)=y_{\sigma_k^t}(k)=C_{\sigma_k^t}x(k)+v_{\sigma_k^t}(k),\forall k\in\{1,\dots,t-1\}.$$

For each $k\leq t$ for $t\leq N$ and each $\sigma^t\in\mathbb{M}^t$, let $\hat{\Sigma}_k^{\sigma^t}$ be the covariance matrix of the optimal estimate of $x(k)$ given the measurements $\{y(0),\dots,y(k-1)\}$. By a standard result of linear estimation theory, the Kalman filter is the minimum mean square error estimator for the system considered, and the covariance of the system evolves according to the Riccati recursion,

$$\begin{aligned}\hat{\Sigma}_{k+1}^{\sigma^t}=&A(k)\hat{\Sigma}_k^{\sigma^t}A(k)^{\mathrm{T}}+\Sigma_w-\\&A(k)\hat{\Sigma}_k^{\sigma^t}C_{\sigma_k^t}^{\mathrm{T}}\Big(C_{\sigma_k^t}\hat{\Sigma}_k^{\sigma^t}C_{\sigma_k^t}^{\mathrm{T}}+\Sigma_{v_{\sigma_k^t}}\Big)^{-1}C_{\sigma_k^t}\hat{\Sigma}_k^{\sigma^t}A(k)^{\mathrm{T}}\end{aligned} \tag{4}$$

with initial condition $\hat{\Sigma}_0=\Sigma_0$ and $k\leq t$. Define $V\left(\sigma^t\right):\mathbb{M}^t\rightarrow\mathbb{R}_+$ as the accrued cost of the weighted trace of the estimation error covariance matrix at each time-step,

$$V(\sigma^t)=\sum_{k=1}^{t}q_k\mathrm{tr}\left(\hat{\Sigma}_k^{\sigma^t}\right)=\sum_{k=1}^{t}q_k\sum_{j=1}^{n}e_j^T\hat{\Sigma}_k^{\sigma^t}e_j \tag{5}$$

where q_k is a non-negative scalar weighting factor for each time-step, and $e_j\in\mathbb{R}^n$ is a unit vector with a one in the j^{th} position and zeros elsewhere. The use of this objective function also enables the representation of the estimation error at the final time-step by setting $q_k=0,\;\forall k\neq N$. Formally, the objective of the problem is,

$$\underset{\sigma^N\in\mathbb{M}^N}{\text{minimize}}\;V\left(\sigma^N\right). \tag{6}$$

III. Properties of the Objective Function

Let $\mathcal{A}$ denote the *positive semidefinite cone*, which is the set of all symmetric positive semidefinite matrices. Similar to [14], a *Riccati Mapping*, $\phi_i:\mathcal{A}\rightarrow\mathcal{A}$ can be defined, which maps the current covariance matrix under the measurement from sensor $i\in\mathbb{M}$ to the next covariance matrix of the estimate,

$$\begin{aligned}\phi_i(\hat{\Sigma}_k)=&A(k)\hat{\Sigma}_kA(k)^{\mathrm{T}}-\\&A(k)\hat{\Sigma}_kC_i^{\mathrm{T}}\Big(C_i\hat{\Sigma}_kC_i^{\mathrm{T}}+\Sigma_{v_i}\Big)^{-1}C_i\hat{\Sigma}_kA(k)^{\mathrm{T}}+\Sigma_w.\end{aligned} \tag{7}$$

Definition 1 (Switched Riccati Mapping): *The mapping $\phi_{\mathbb{M}}:2^{\mathcal{A}}\rightarrow2^{\mathcal{A}}$, defined by:*

$$\phi_{\mathbb{M}}\left(\mathcal{S}_k\right)=\left\{\phi_i\left(\hat{\Sigma}_k\right):\forall i\in\mathbb{M},\;\forall\hat{\Sigma}_k\in\mathcal{S}_k,\mathcal{S}_k\in2^{\mathcal{A}}\right\}.$$

The switched Riccati mapping maps the set of positive semidefinite matrices to another set of positive semidefinite matrices by mapping each matrix in $\mathcal{S}_k$ through each possible sensor measurement.

Definition 2 (Switched Riccati Sets): *The sequence of sets $\{\mathcal{S}_k\}_{k=0}^N$ that can be generated iteratively through $\mathcal{S}_{k+1}=\phi_{\mathbb{M}}\left(\mathcal{S}_k\right)$ with an initial condition of $\mathcal{S}_0=\{\Sigma_0\}$ are called the switched Riccati sets.*

The switched Riccati sets grow exponentially in size from the singleton set $\{\Sigma_0\}$ to the set $\mathcal{S}_N$ consisting of M^N positive semidefinite matrices. The switched Riccati sets express the covariance of the estimate at every time-step under every possible sensor schedule.

Let $\mathcal{S}_k(i)$ be the i^{th} element of the set $\mathcal{S}_k$, $\kappa(\mathcal{S}_k(i))\in\mathbb{M}^k$ be the ordered sensor schedule corresponding to the covariance estimate of the state $\mathcal{S}_k(i)$ and κ^* be the optimal sensor schedule for the problem. Figure 1 depicts the search tree for the sensor schedule. The tree grows exponentially with each time-step, requiring careful development of solutions with computationally tractable algorithms.

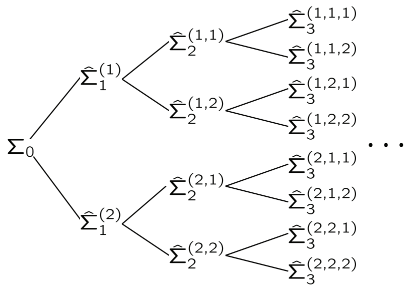

Fig. 1. The search tree for the sensor scheduling problem which is the enumeration of all possible sensor schedules and the covariance of the estimate at each time-step.

An important property of the Kalman filter is given in the following lemma.

Lemma 1: *Suppose $\hat{\bar{\Sigma}}_k$ and $\hat{\Sigma}_k$ are the Kalman Filter covariance estimates from the initial conditions $\bar{\Sigma}_0$ and Σ_0, respectively. If $\bar{\Sigma}_0 \succeq \Sigma_0$ then $\hat{\bar{\Sigma}}_k \succeq \hat{\Sigma}_k$, $\forall k \geq 0$.*

Proof: The proof of this lemma is provided in [15]. ■

It states that systems starting with a larger initial covariance, in the positive semidefinite sense, will yield larger covariances at all future time-steps. This result is important because it provides insight into how to reduce the complexity of the problem. Using the results from Lemma 1, Theorem 1 provides a condition on the branches of this search tree which can be pruned with respect to the objective considered for the sensor scheduling problem.

Theorem 1: *If $\hat{\Sigma}_k^{\bar{\sigma}^k} \succeq \hat{\Sigma}_k^{\tilde{\sigma}^k}$ and $V(\bar{\sigma}^k) \geq V(\tilde{\sigma}^k)$ then the branch defined by $\bar{\sigma}^k$ and all of its descendants can be pruned without eliminating the optimal solution from the search tree.*

Proof: From Lemma 1, $\hat{\Sigma}_{k+t}^{(\bar{\sigma}^k,\sigma^t)} \succeq \hat{\Sigma}_{k+t}^{(\tilde{\sigma}^k,\sigma^t)}$, $\forall t \in \{1,\dots,N-k\}$ and $\forall \sigma^t \in \mathbb{M}^t$. Therefore,

$$\sum_{t=1}^{N-k}\sum_{j=1}^{n} e_j^T \hat{\Sigma}_{k+t}^{((\bar{\sigma}^k,\sigma^t))} e_j \geq \sum_{t=1}^{N-k}\sum_{j=1}^{n} e_j^T \hat{\Sigma}_{k+t}^{((\tilde{\sigma}^k,\sigma^t))} e_j$$

Consequently, if $V(\bar{\sigma}^k) \geq V(\tilde{\sigma}^k)$ then,

$$V(\bar{\sigma}^k) + \sum_{t=1}^{N-k} q_{k+t}\mathrm{tr}\hat{\Sigma}_{k+t}^{(\bar{\sigma}^k,\sigma^t)} \geq c(\tilde{\sigma}^k) + \sum_{t=1}^{N-k} q_{k+t}\mathrm{tr}\hat{\Sigma}_{k+t}^{(\tilde{\sigma}^k,\sigma^t)}.$$

Therefore the branch defined by $\bar{\sigma}^k$ and its descendants can be eliminated because it will not have a smaller cost than any sensor schedule starting with $\tilde{\sigma}^k$. ■

IV. Description of Algorithms

Using the properties of the objective function and search tree, two efficient algorithms will be developed. The first method will employ pruning of the search tree via Theorem 1 to provide the optimal solution. The second method will reduce the complexity of the problem even more but will only yield a suboptimal solution.

A. Optimal Solution

The method of enumerating all possible sensor schedules is only tractable for relatively short time horizons, but through efficient pruning of the search tree, larger time horizon solutions are possible. Theorem 1 can be used to define a condition that characterizes the redundancy of a branch with respect to other branches. This is presented in the following lemma.

Lemma 2: *$\bar{\Sigma}$ is a redundant matrix with respect to the set of matrices $\mathcal{S}_k$ if $\exists i \in \{1,\dots,|\mathcal{S}_k|\}$ such that $\bar{\Sigma} \succeq \mathcal{S}_k(i)$ and $V\left(\kappa\left(\bar{\Sigma}\right)\right) \geq V\left(\kappa\left(\mathcal{S}_k(i)\right)\right)$.*

Let an equivalent subset of the search tree be defined as one that still contains the optimal sensor schedule. In computing the switched Riccati sets, Lemma 2 can be applied to calculate an equivalent subset of $\mathcal{S}_k$, $\forall k \in \{1,\dots,N\}$, which is outlined in Algorithm 1. The first step is to sort the set in ascending order based upon the current cost of the branches. Sorting the branches first will reduce the complexity of the algorithm by requiring fewer evaluations of Lemma 2. The equivalent subset is initialized to the current best branch. Next, each entry in $\mathcal{S}_k$ is tested with the current equivalent subset, $\mathcal{S}_k^{(i-1)}$, to determine if it can be eliminated. If not, then it is appended to the current subset.

Algorithm 1 Computation of the Equivalent Subsets
```
1: sort $\mathcal{S}_k$ in ascending order such that $V(\kappa(\mathcal{S}_k(i))) \leq V(\kappa(\mathcal{S}_k(i+1)))$, $\forall i \in \{1,\dots,|\mathcal{S}_k|-1\}$.
2: $\mathcal{S}_k^{(i)} = \{\mathcal{S}_k(1)\}$
3: for $i$ = 2,...,$|\mathcal{S}_k|$ do
4:     if $\mathcal{S}_k(i)$ satisfies Lemma 2 with $\mathcal{S}_k^{(i-1)}$ then
5:         $\mathcal{S}_k^{(i)} = \mathcal{S}_k^{(i-1)}$
6:     else
7:         $\mathcal{S}_k^{(i)} = \mathcal{S}_k^{(i-1)} \cup \mathcal{S}_k(i)$
8:     end if
9: end for
```

An efficient method for computing the optimal sensor schedule which uses the proposed pruning technique is stated in Algorithm 2. The procedure first initializes the switched Riccati set to the a priori covariance of the initial state. Then, for each time-step it computes the switched Riccati mapping and calculates the equivalent subset with Algorithm 1. Once the tree is fully built, the optimal sensor schedule is determined.

Algorithm 2 Sensor Scheduling for a Finite Horizon
```
1: $\mathcal{S}_0 = \{\Sigma_0\}$
2: for $k$ = 1,...,$N$ do
3:     $\mathcal{S}_k = \phi_{\mathbb{M}}(\mathcal{S}_{k-1})$
4:     Perform Algorithm 1 with $\mathcal{S}_k$
5: end for
6: $\kappa^* = \underset{j\in\{1,\dots,|S_N|\}}{\arg\min} V(\kappa(S_N(j)))$
```

The complexity of the algorithm will be demonstrated through an example. Consider the system defined by Eqn. (1)

with

$$A = \begin{bmatrix} -0.09 & -1.21 \\ 0.47 & 1.37 \end{bmatrix}, \quad \Sigma_w = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$C_1 = \begin{bmatrix} 0.70 & 0.64 \end{bmatrix}, \qquad \Sigma_{v_1} = 0.13,$$
$$C_2 = \begin{bmatrix} 0.78 & 0.72 \end{bmatrix}, \qquad \Sigma_{v_2} = 0.82,$$
$$C_3 = \begin{bmatrix} 0.93 & 0.19 \end{bmatrix}, \qquad \Sigma_{v_3} = 0.98$$

and a schedule horizon of $N = 50$ with $q_k = 1, \forall k$. Figure 2 compares the number of branches required to obtain the optimal solution to the sensor scheduling problem for the brute force enumeration of all the branches versus Algorithm 2 which also provides the optimal solution but prunes redundant branches. At the final time-step, there are 10^{26} branches in the whole tree, but only 295 are required for the pruning algorithm.

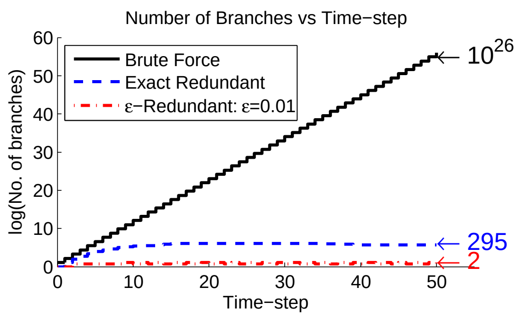

Fig. 2. Comparison of the number of branches at each time-step for the brute force enumeration, the exact redundant elimination and the numerical redundant algorithm with $\epsilon = 0.01$.

Even though the optimal solution prunes a large number of branches, the growth of the search tree may still become prohibitive for some problems. Therefore, an approximate solution may be desired.

B. Suboptimal Solution

The following section describes an algorithm which approximates the search tree by pruning branches which are numerically redundant. Similar to Lemma 2, the following lemma provides a condition for testing the ϵ-redundancy of a matrix.

Lemma 3: *$\bar{\Sigma}$ is a ϵ-redundant matrix with respect to the set of matrices $\mathcal{S}_k$ if $\exists i \in \{1, \ldots, |\mathcal{S}_k|\}$ such that $\bar{\Sigma} + \epsilon I_n \succeq \mathcal{S}_k(i)$ and $V\left(\kappa\left(\bar{\Sigma}\right)\right) \geq V\left(\kappa\left(\mathcal{S}_k(i)\right)\right)$.*

Figure 3 illustrates the premise behind ϵ-redundancy with respect to Lemma 3. In this example, $S_k(i)$ cannot strictly eliminate $\bar{\Sigma}$ but for some ϵ, $S_k(i)$ is completely contained within $\bar{\Sigma} + \epsilon I$. Consequently, for that ϵ if $V\left(\kappa\left(\bar{\Sigma}\right)\right) \geq V\left(\kappa\left(\mathcal{S}_k(i)\right)\right)$ then $\bar{\Sigma}$ can be eliminated. Using the ϵ-approximation further reduces the number branches in the search tree and the complexity of the problem. Consequently, this will enable the solution of problems that are intractable for the optimal algorithm.

From Theorem 1, only the solution from the branch $\bar{\Sigma} + \epsilon I_n$ can be guaranteed to be worse than the solution from the branch that eliminated it. Therefore, the optimal solution can no longer be guaranteed and the distance within optimality cannot be bounded as well. However, simulation results suggest that for small ϵ the solution is close to optimal.

To efficiently compute the ϵ-equivalent subset, Algorithm 1 can be modified to use Lemma 3 instead of Lemma 2 and is denoted as $Algo_{NR}(\epsilon)$. Finally, to determine the ϵ-approximate solution of the sensor scheduling problem, Algorithm 2 can be modified by substituting $Algo_{NR}(\epsilon)$ for Algorithm 1 and is referred to as $Algo_S(\epsilon)$.

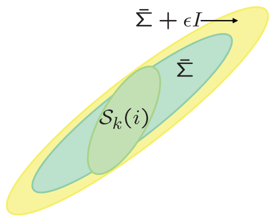

Fig. 3. Example covariances that demonstrate the concept of ϵ-redundancy.

Figure 2 compares the number of branches for the optimal and suboptimal algorithm with $\epsilon = 0.01$. The suboptimal algorithm significantly reduced the complexity of the problem requiring only 2 branches compared with 295 for the optimal algorithm. An example of the ϵ-approximate solution is illustrated for the system defined by Eqn. (1) with

$$A = \begin{bmatrix} -0.6 & 0.8 & 0.5 \\ -0.1 & 1.5 & -1.1 \\ 1.1 & 0.4 & -0.2 \end{bmatrix}, \quad \Sigma_w = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$
$$C_1 = \begin{bmatrix} 0.75 & -0.2 & -0.65 \end{bmatrix}, \qquad \Sigma_{v_1} = 0.53,$$
$$C_2 = \begin{bmatrix} 0.35 & 0.85 & 0.35 \end{bmatrix}, \qquad \Sigma_{v_2} = 0.8,$$
$$C_3 = \begin{bmatrix} 0.2 & -0.65 & 1.25 \end{bmatrix}, \qquad \Sigma_{v_3} = 0.2,$$
$$C_4 = \begin{bmatrix} 0.7 & 0.5 & 0.5 \end{bmatrix}, \qquad \Sigma_{v_4} = 0.5$$

and $q_k = 1, \forall k$. In this example, the horizon length is $N = 50$ which results in 10^{30} branches for the naive brute force search. Figure 4 shows the resulting sensor schedule for the suboptimal algorithm. This solution was the same for $\epsilon = \{0.01, 0.1, 0.2, 0.5\}$ with objective function value of 850.57. After the initial transients settle out around time-step 9, the sensor schedule is periodic with the sequence $\{1, 2, 3, 4, 1, 4, 2\}$. Figure 5 shows the number of branches in the search tree per time-step for $\epsilon = \{0.01, 0.1, 0.2, 0.5\}$, which converge to 624, 95, 50 and 24, respectively. Typically, the number of branches in the search tree converge to a small number for large ϵ.

Table I compares the performance of different ϵ for 500 random cases in which there are $M = 3$ sensors and $n = 4$ states with a planning horizon of length $N = 50$. Each

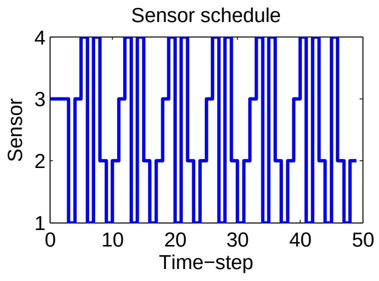

Fig. 4. Suboptimal sensor schedule for $\epsilon = \{0.01, 0.1, 0.2, 0.5\}$.

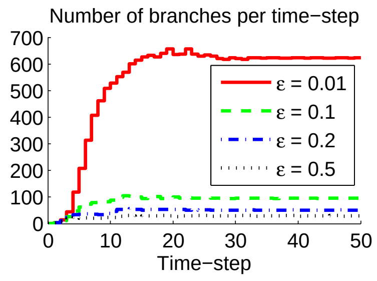

Fig. 5. Number of matrices per time-step for several suboptimal solutions $\epsilon = \{0.01, 0.1, 0.2, 0.5\}$.

row represents the percentage of cases in which for that ϵ it obtained a strictly larger objective cost than another ϵ. For example, in 53.9% of the cases $Algo_S(5.0)$ performed worse than $Algo_S(0.1)$, and for 0.6% of the cases $Algo_S(0.1)$ performed worse than $Algo_S(0.5)$. In general, smaller ϵ will generate a better sensor schedule with a smaller objective function than larger ϵ, but in rare cases larger ϵ can obtain a better solution. Consider the following example, which refers to Figure 1, to explain this uncommon occurrence. Let the branch $(1, 2, 1)$ be the optimal solution in this problem. For $\epsilon = 0.1$, at time-step 2 the branch $(2, 1)$ will eliminate $(1, 2)$. However for $\epsilon = 1.0$, at time-step 1 the branch (1) will eliminate (2) since it has a larger approximation, and therefore $(1, 2)$ will not be eliminated at the next time-step. Consequently, the larger ϵ will yield a better solution than the smaller ϵ.

TABLE I

EACH ROW REPRESENTS THE PERCENTAGE OF CASES IN WHICH FOR THAT ϵ IT OBTAINED A STRICTLY LARGER OBJECTIVE COST THAN THE ϵ IN THE COLUMN.

ϵ	0.1	0.2	0.5	1.0	2.0	3.0	5.0
0.1	-	0.2	0.6	0.6	0.2	0.2	0.0
0.2	3.7	-	0.8	0.8	0.8	1.0	0.2
0.5	8.8	5.5	-	0.8	1.8	1.4	0.6
1.0	16.0	12.7	9.6	-	1.8	1.6	1.2
2.0	32.8	30.9	29.3	23.6	-	2.5	1.3
3.0	42.2	40.2	39.6	35.5	21.3	-	4.3
5.0	53.9	52.9	52.3	49.2	39.1	29.9	-

C. Performance

To characterize the performance of the suboptimal algorithm, 100 random instances were performed with $M = 3$ sensors, state dimension $n = 4$, $q_k = 1$, $\forall k \in \{1, \ldots, N\}$, and a horizon of length $N = 14$. In generating the random systems, the pair (A, C_i), $\forall i \in \mathbb{M}$, were restricted to be unobservable, with the exception that if all the sensors are used at once then the system is fully observable. For each problem, both the optimal solution and the suboptimal solutions over $\epsilon = \{0.1, 0.2, \ldots, 1.0\}$ were calculated. Figure 6(a) displays the percentage of the solutions that are optimal for each ϵ. As ϵ is increased there is a slow decay of the number that are optimal. Figure 6(b) displays the mean and maximum percentage of the final cost over the optimal solution for each ϵ. For all ϵ, the solution is well within 0.5% of the optimal objective function value for most of the instances and is closer to optimal as ϵ is decreased. Figure 6(c) shows the number of branches in the search tree at the final time-step. As ϵ increases, fewer branches are needed to represent the search tree, and even an $\epsilon = 0.1$ requires on average four orders of magnitude less branches than brute force enumeration. As illustrated in the figure, the general trend for both the mean and maximum values is an exponential decay as ϵ increases.

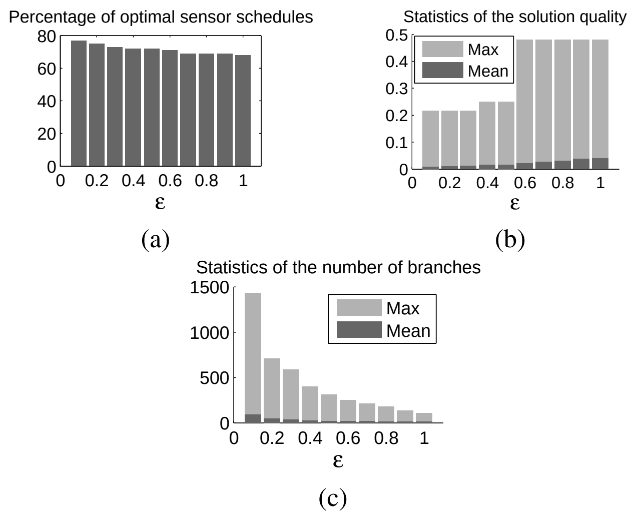

Fig. 6. Comparison of the performance of the suboptimal algorithm for different ϵ. (a) The percentage of solutions for the suboptimal algorithm that are the optimal solution. (b) Mean and maximum percentage of the objective function over the optimal solution for each ϵ. (c) Comparison of the mean and maximum number of branches in the search tree at the final time-step for each ϵ

V. HIERARCHICAL ALGORITHM

For larger ϵ, a fewer number of matrices are needed to characterize the objective function and therefore the sensor schedule can be quickly computed. Another important aspect of the ϵ-approximate algorithm is that in general, the smaller the ϵ the closer the solution is to optimal since more branches are being explored. Therefore, it would be desirable to be able to combine the benefits from both the larger and smaller ϵ.

To this end, a hierarchical algorithm can be devised which uses an upperbound on the objective function, $\bar{V}$, acquired from the larger ϵ to prune branches when computing the solution with smaller ϵ. The upperbound can be used in two different ways to prune branches for the smaller ϵ. A branch can be pruned if the current value of the objective function is larger than the upperbound or if a lowerbound on the objective function for that branch is larger than the upperbound. A lowerbound for the future cost can be obtained through the use of *all* the sensors to determine an estimate of the state. This is indeed a lowerbound because the Kalman filter is the minimum mean square estimator for a linear, Gaussian system and by removing sensor measurements the estimate cannot be improved.

Let $\underline{V}(\cdot)$ define a lowerbound of the objective function for the remaining time-steps. The lowerbound, $\underline{V}\left(\hat{\Sigma}_k\right)$ can be calculated through iterating the same Riccati recursion in Eqn. 4 for $N-k$ times with $C = \left[C_1^T, \ldots, C_M^T\right]^T$ and $\Sigma_v = \operatorname{diag}\left(\Sigma_{v_1}, \ldots, \Sigma_{v_M}\right)$ where $\operatorname{diag}(\cdot)$ places the elements along the diagonal.

Lemma 4: *$\bar{\Sigma}$ is a ϵ-redundant matrix with respect to the set of matrices $\mathcal{S}_k$ if $\exists i \in \{1, \ldots, |\mathcal{S}_k|\}$ such that $\bar{\Sigma} + \epsilon I_n \succeq \mathcal{S}_k(i)$ and $V\left(\kappa\left(\bar{\Sigma}\right)\right) \geq V\left(\kappa\left(\mathcal{S}_k(i)\right)\right)$ or $V\left(\kappa\left(\bar{\Sigma}\right)\right) + \underline{V}\left(\mathcal{S}_k(i)\right) \geq \bar{V}$.*

Denote $Algo_{hes}\left(\epsilon, \bar{V}\right)$ a modified version of Algorithm 1 with Lemma 2 replaced with Lemma 4 to compute the ϵ-equivalent subset with additional pruning based upon an upperbound on the objective function. Also, denote $Algo_H\left(\epsilon, \bar{V}\right)$ a modified version of Algorithm 2 with Algorithm 1 replaced with $Algo_{hes}\left(\epsilon, \bar{V}\right)$. Finally, the hierarchical sensor scheduling algorithm can be stated in Algorithm 3. The first step is to compute a sensor schedule for a relatively large ϵ to obtain an upperbound on the objective function. Next, the hierarchical algorithm, $Algo_H(\cdot)$, is performed with a smaller ϵ and the upperbound from the first step. The last step is to choose the sensor schedule that has the smallest value of the objective function. If a better solution is required, another iteration of the hierarchical algorithm, $Algo_H(\cdot)$, can be performed with an even smaller ϵ.

Algorithm 3 Hierarchical Sensor Scheduling for a Finite Horizon

```
1: $\hat{\kappa}^*$ = Perform $Algo_S(\hat{\epsilon})$
2: $\bar{V} = V\left(\hat{\kappa}^*\right)$
3: $\check{\kappa}^*$ = Perform $Algo_H\left(\check{\epsilon}, \bar{V}\right)$
4: $\kappa^* = \underset{\kappa \in \{\hat{\kappa}^*,\, \check{\kappa}^*\}}{\arg\min}\ V(\kappa)$
```

Figure 7 shows the performance of the hierarchical method for 100 difficult[1] random cases with state dimension $n = 4$, $M = 3$ sensors, $q_k = 1\ \forall k \in \{1, \ldots, N\}$ and a horizon of $N = 50$. As before, the pair (A, C_i), $\forall i \in \mathbb{M}$, were restricted to be unobservable, but if all the sensors are used at once then the system is fully observable. For the hierarchical algorithm, an $\epsilon = 1.0$, which requires on average only 40 branches, is used as the bounding solution to prune the search tree for either $\epsilon = 0.1$ or $\epsilon = 0.5$. As expected, for smaller ϵ the hierarchical algorithm is able to prune more branches. The average number of branches required for the hierarchical method for $\epsilon = 0.5$ is 15 compared with 73 for the non-hierarchical method, and for $\epsilon = 0.1$ is 23 compared with 327 for the non-hierarchical method. Also note that the algorithm is only able to prune branches after time-step 7 because the lowerbound on the objective function is not a good representation of the cost of the branch in the early stages of building the tree.

[1]Typically, if the larger ϵ needed a significant number of branches to represent the search tree, then the sensor scheduling problem was deemed difficult. A difficult case was required to have at least 15 branches at the final time-step for $\epsilon = 1.0$ to represent the search tree.

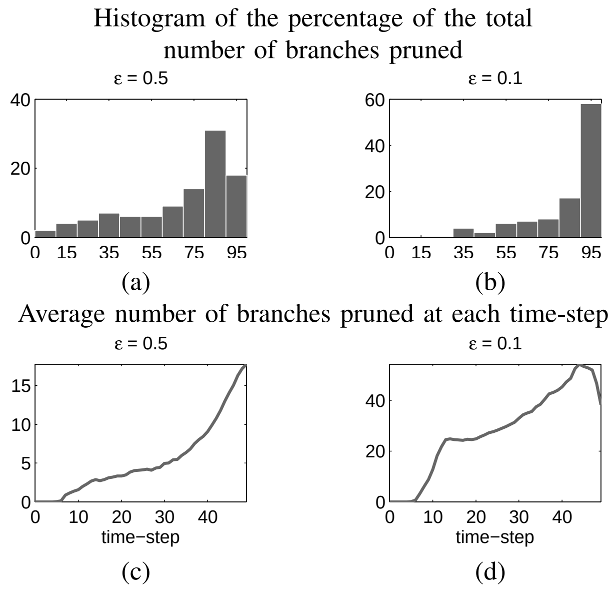

Fig. 7. (a)-(b) Histograms showing the percentage of the original number of branches pruned by the hierarchical method for $\epsilon = 0.5$ and $\epsilon = 0.1$, respectively. (c)-(d) Plots of the average number of branches pruned at each time-step for $\epsilon = 0.5$ and $\epsilon = 0.1$, respectively.

VI. Conclusions

To solve the sensor scheduling problem, a condition on when a partial sensor schedule cannot become part of the optimal schedule was developed. Using this condition, two algorithms were devised, which provide the optimal and suboptimal solutions, to prune the search tree to enable the solution of larger systems and longer time horizons. The algorithms trade-off between the quality of the solution and the complexity of the problem. A hierarchical algorithm was also developed to reduce the complexity of the problem while maintaining the quality of the solution.

An area of interest for future work is to extend these methods to consider the case when the sensors depend on the state of the system. Another area of interest is modifying the objective function to include a direct measure of the power consumption of each sensor. Lastly, to develop a tighter lower bound for pruning branches in the hierarchical algorithm to improve its performance.

References

[1] L. Meier III, J. Peschon, and R. M. Dressler, "Optimal control of measurement subsystems," *IEEE Transactions on Automatic Control*, vol. 12, pp. 528–536, October 1967.

[2] P. Alriksson and A. Rantzer, "Sub-optimal sensor scheduling with error bounds," in *Proceedings of the 16th IFAC World Congress*, (Prague, Czech Republic), July 2005.

[3] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *To appear, IEEE Transactions on Signal Processing*, 2008.

[4] V. Gupta, T. Chung, B. Hassibi, and R. M. Murray, "Sensor Scheduling Algorithms Requiring Limited Computation," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[5] V. Gupta, T. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, 2004.

[6] Y. He and K. Chung, "Sensor Scheduling for Target Tracking in Sensor Networks," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, (Atlantis, Paradise Island, Bahamas), December 2004.

[7] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, pp. 407–427, March 1999.

[8] C. Tomlin, J. Lygeros, and S. Sastry, "A game theoretic approach to controller design for hybrid systems," *In Proceedings of IEEE*, vol. 88, pp. 949–969, July 2000.

[9] F. Borrelli, M. Baotic, A. Bemporad, and M. Morari, "Dynamic programming for constrained optimal control of discrete-time linear hybrid systems," *Automatica*, vol. 41, pp. 1709–1721, Oct. 2005.

[10] B. Lincoln and A. Rantzer, "Relaxed optimal control of piecewise linear systems," in *Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems*, June 2003.

[11] M. Branicky and G. Zhang, "Solving hybrid control problems: Level sets and behavioral programming," in *In Proc. American Contr. Conf.*, June 2000.

[12] A. Hassibi and S. Boyd, "Quadratic stabilization and control of piecewise-linear systems," in *In Proc. American Contr. Conf.*, June 1998.

[13] W. Zhang, J. Hu, and A. Abate, "Switched LQR problem in discrete time: Theory and algorithms," *submitted to IEEE Transactions on Automatic Control*, 2008.

[14] W. Zhang and J. Hu, "On the Value Functions of the Optimal Quadratic Regulation Problem for Discrete-Time Switched Linear Systems," in *Proceedings of IEEE International Conference on Decision and Control*, (Cancun, Mexico), December 2008.

[15] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Englewood Cliffs, NJ, USA: Prentice-Hall, Inc., 1986.